REMARKS

As a preliminary matter, Applicant acknowledges and appreciates the Examiner's withdrawal of the previous 35 USC § 112, first paragraph rejections of claims 9 – 11 and 22 – 25.

Further, Applicants wish to thank Examiner Wang for his courtesy in granting a telephone interview to discuss the application. An interview was conducted July 29, 2008 between Examiner Wang and Applicants' attorney Arik Ranson. Claims 1 and 27 were discussed, as was reference Goodnow. Though no final resolution was reached, Applicants believe the Examiner has a better understanding of key differences between Goodnow and embodiments of the present invention.

Claims 1 – 8, 12 – 17, 26, 37 – 40, 42 – 48, 50, and 52 stand rejected under 35 USC § 102(b) as being anticipated by US Patent No. 5,699,507 to Goodnow. Applicant has amended independent claims 1, 27, and 35 to more clearly define, among other things, that each of the plurality of intervals of execution in claims 1 and 35, and each of the arbitrary sections of execution in claim 27, is defined by continuous instructions of the code in program execution order run over a defined amount of time, wherein the amount of time is defined by at least one of a predefined time interval, a predefined number of instructions that are to be run, and a time having a length based on a predefined metric. This amendment does not introduce new matter, as clear support for the amendment can be found in the present application as filed. For example, please see page 11, lines 3-12.

As applied to the claims as amended, Applicant respectfully traverses the rejection. Goodnow fails to teach or suggest at least identifying a behavior of a computer program over each of a plurality of intervals of execution based on a tracked statistic for a program component, or comparing at least one identified behavior for at least one interval of execution to another interval of execution to determine similarity between the intervals of execution, where each of the plurality of intervals of execution is defined by continuous instructions of the code in program execution order run over a defined amount of time, and wherein the amount of time is defined by at least one of a predefined time interval, a predefined number of instructions that are to be run, and a time having a length based on a predefined metric.

Goodnow, instead, identifies behavior and compares identifies behavior of code segments, which are always defined by specific lines of code, not by time. The present Office Action asserts regarding Goodnow, "The code segments may be identified as two functions within either the same program or different programs or any other defined designations such as, but not limited to blocks, Lvalues or statements." However, these alternative ways of identifying or labeling the code segments do not change the fact that the code segments are always defined by, and restricted to, particular lines of code. Code segments are defined by specific lines of code (C4, L42-45; C4, L62-65). Functions are defined by specific lines of code (Table 1; C5, L9-15; C10, L17 ("A function corresponds to a particular collection of blocks")). Blocks are defined by specific lines of code (C9, L13-14). Lvalues identify specific lines of code (an assigned location). Statements identify specific lines of code (by definition). Even when

17

Goodnow relies on a code extractor to obtain code segments, as described in the related patent 5,574,837 incorporated therein by reference, these code segments are extracted (generated) and sent as specific lines of code. Nothing in column 4, or anywhere else in Goodnow (including the patent incorporated therein by reference), contradicts this.

To define intervals of execution by time in Goodnow would undermine the very nature of Goodnow's invention, because it would not allow evaluation and comparison of specific, identifiable pieces of code. The purpose of Goodnow's invention is to test particular lines of code so that they can be incorporated into one or more other programs. If the code segments are not defined by particular lines of code, there is nothing in particular to test. All of the testing described in Goodnow is performed with respect to code segments, not intervals of execution defined by time.

Thus, Goodnow's invention does not teach or suggest identifying behavior over each of a plurality of intervals of execution of arbitrary sections of execution defined by time, or comparing at least one identified behavior for at least one interval of execution to another interval of execution to determine similarity between the intervals of execution, as defined in amended independent claim 1 and dependent claims 2-8, 12-17, 26, 43-48, 50, and 52. Applicants thus respectfully request reconsideration and withdrawal of the rejection.

Claims 9-11, 22-25, 27-33, 35, and 36 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Goodnow in view of a publication by present inventor Calder, "Time Varying Behavior of Programs" ("Calder"). Applicants respectfully traverse the rejection.

Regarding dependent claims 9-11 and 22-25, Applicants respectfully traverse the rejection for at least the reason that Goodnow fails to teach or suggest at least identifying a behavior of a computer program over each of a plurality of intervals of execution based on a tracked statistic for a program component, or comparing at least one identified behavior for at least one interval of execution to another interval of execution to determine similarity between the intervals of execution, where each of the plurality of intervals of execution is defined by continuous instructions of the code in program execution order run over a defined amount of time, and wherein the amount of time is defined by at least one of a predefined time interval, a predefined number of instructions that are to be run, and a time having a length based on a predefined metric, as defined in independent claim 1. Applicants' arguments regarding independent claim 1 and Goodnow are incorporated herein.

Regarding independent claim 27 and dependent claims 28-33 and 35, Applicant respectfully traverses the rejection for at least the reason that Goodnow fails to teach or suggest at least identifying a behavior of a hardware-independent metric within at least one arbitrary section of execution, and classifying each of the at least one arbitrary section of execution according to the identified behavior into clusters of behavior, where each of the arbitrary sections of execution is defined by continuous instructions of the code in program execution order run over a defined amount of time, and wherein the amount of time is defined by at least one of a predefined time interval, a predefined number of instructions that are to be run, and a time having a length based on a predefined metric. For similar reasons as stated above regarding claim 1 and Goodnow,

19

the reference fails to teach or suggest at least identifying behavior of a hardware-independent metric within at least one arbitrary section of execution. Further, any clustering performed by Goodnow is with respect to functions that, again, are defined by specific lines of code, not by time (for example, see C13, L18, describing a cluster interface for functions f_0, f_1, f_2, f_3, and f_4, all defined by particular lines of code).

With respect to independent claim 35 and dependent claim 36, Applicants respectfully traverse the rejection for at least the reason that Goodnow fails to teach or suggest at least identifying a behavior of a hardware-independent metric for each of a plurality of intervals of execution, or comparing the identified behavior of each of the plurality of intervals to identified target behavior over full execution of at least a portion of a computer program, where each of the plurality of intervals of execution is defined by continuous instructions of the code in program execution order run over a defined amount of time, and wherein the amount of time is defined by at least one of a predefined time interval, a predefined number of instructions that are to be run, and a time having a length based on a predefined metric. Goodnow fails to teach or suggest identifying a behavior of a hardware-independent metric for each of a plurality of intervals of execution for similar reasons as that stated above regarding claim 1. Additionally, for similar reasons, Goodnow fails to teach or suggest comparing identified behavior of each of the plurality of intervals to identified target behavior over full execution of at least a portion of a computer program.

Further, with respect to claims 9-11, 22-25, 27-33, 35, and 36, secondary reference Calder fails to remedy the deficiencies of Goodnow regarding the claims.

20

Calder is cited for simulating execution over intervals of 10 million or 100 million. However, Calder does not teach or suggest tracking a statistic for a program component for intervals of execution, as defined in independent claim 1, identifying behavior of a hardware-independent metric for arbitrary sections of execution, as defined in independent claim 27, or identifying behavior of a hardware-independent metric for each of a plurality of intervals of execution, as defined in independent claim 35. Instead, Calder specifically teaches identifying behavior based on hardware benchmarks (for example, see page 4, describing benchmark data such as percent Register Update Unit (RUU) occupancy, cache miss rate, branch prediction miss rate, address prediction miss rate, and value prediction miss rate).

By contrast, embodiments of the present invention allow behavior intervals of execution or arbitrary sections of execution (as defined in the claims) to be identified and compared, either to one another or to an overall behavior of a program, based on a hardware-independent metric, such as the code itself (e.g., a program component). More particular embodiments of the present invention (for example, as defined in one or more dependent claims) provide that execution of a computer program can be classified into phases based on such behavior, or compared to the behavior of a program (or a portion thereof) as a whole.

Note, however, that a program component, such as but not limited to a basic block, is an example of a metric to evaluate behavior of an interval of execution or arbitrary section of execution in embodiments of the present invention, but this should not be confused with how the interval of execution or arbitrary section of execution itself

21

is <u>defined</u>. By contrast, Goodnow uses a specific block of instructions to define its "interval", and Calder uses a hardware metric to evaluate intervals of instructions.

For at least these reasons, Applicants respectfully submit that claims 9-11, 22-25, 27-33, 35, and 36 are allowable over the references of record, including Goodnow and Calder. Applicants thus request reconsideration and withdrawal of the rejection.

Claim 34 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Goodnow in view of Calder, and further in view of Clark. Claims 18-21, 41, 49, 51, and 53 stand rejected over Goodnow in view of Calder, Clark, and Baker. Applicants respectfully traverse the rejections for at least the reasons stated above regarding independent claims 1, 27, and 35, and for at least the additional reason that references Clark and Baker do not remedy the deficiencies of Goodnow and Calder regarding the claims. Applicants thus request reconsideration and withdrawal of the rejection.

For at least the above reasons, Applicants respectfully submit that this application is in condition for allowance, which is respectfully requested. The Examiner is invited to contact Applicants' attorney if an interview would expedite prosecution.

Customer No. 24978
September 24, 2008
300 South Wacker Drive
Suite 2500
Chicago, Illinois 60606
Telephone: (312) 360-0080
Facsimile: (312) 360-9315
P:\DOCS\0321\68199\DF9323.DOC

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By:___/Arik B. Ranson/_____
        Arik B. Ranson
        Registration No. 43,874